## **Cloud Function & Firestore Trigger Challenge**

**Objective:** Demonstrate your understanding of Cloud Functions and Firestore triggers by building a simple system that logs user activity using NodeJs.

**Scenario:** Imagine we're building a platform where users can "like" different articles. We want to keep track of these likes in a separate Firestore collection for analytics purposes.

**Your Task:**

1. **Create a Cloud Function:**
   * This function will be triggered by an HTTP request.
   * It should accept the following data in the request body:
     + userId: The ID of the user who liked the article.
     + articleId: The ID of the article that was liked.
   * The function should then write a new document to a Firestore collection called "article\_likes". This document should contain:
     + userId: The ID of the user who liked the article.
     + articleId: The ID of the article that was liked.
     + timestamp: A timestamp indicating when the like occurred.
   * The function should return a success message upon successful write to Firestore.
2. **Create a Firestore Trigger:**
   * This trigger should be set to activate on the creation of new documents in the "article\_likes" collection.
   * When triggered, the function should:
     + Read the userId and articleId from the newly created document.
     + Write a new document to a Firestore collection called "user\_activity". This document should contain:
       - userId: The ID of the user.
       - activityType: The type of activity, in this case, "article\_liked".
       - articleId: The ID of the liked article.
       - timestamp: A timestamp indicating when the activity was logged.

**Bonus Points:**

* Implement error handling in both the Cloud Function and the Firestore trigger.
* Write unit tests for your Cloud Function.

**Deliverables:**

* The code for your Cloud Function and Firestore trigger.
* Clear instructions on how to deploy and test your code.

**Evaluation Criteria:**

* **Functionality:** Does your code meet all the requirements of the task?
* **Code Quality:** Is your code well-structured, readable, and efficient?
* **Error Handling:** Have you implemented appropriate error handling mechanisms?
* **Testing:** Have you written unit tests to ensure the correctness of your code?
* **Documentation:** Are your instructions clear and easy to follow?